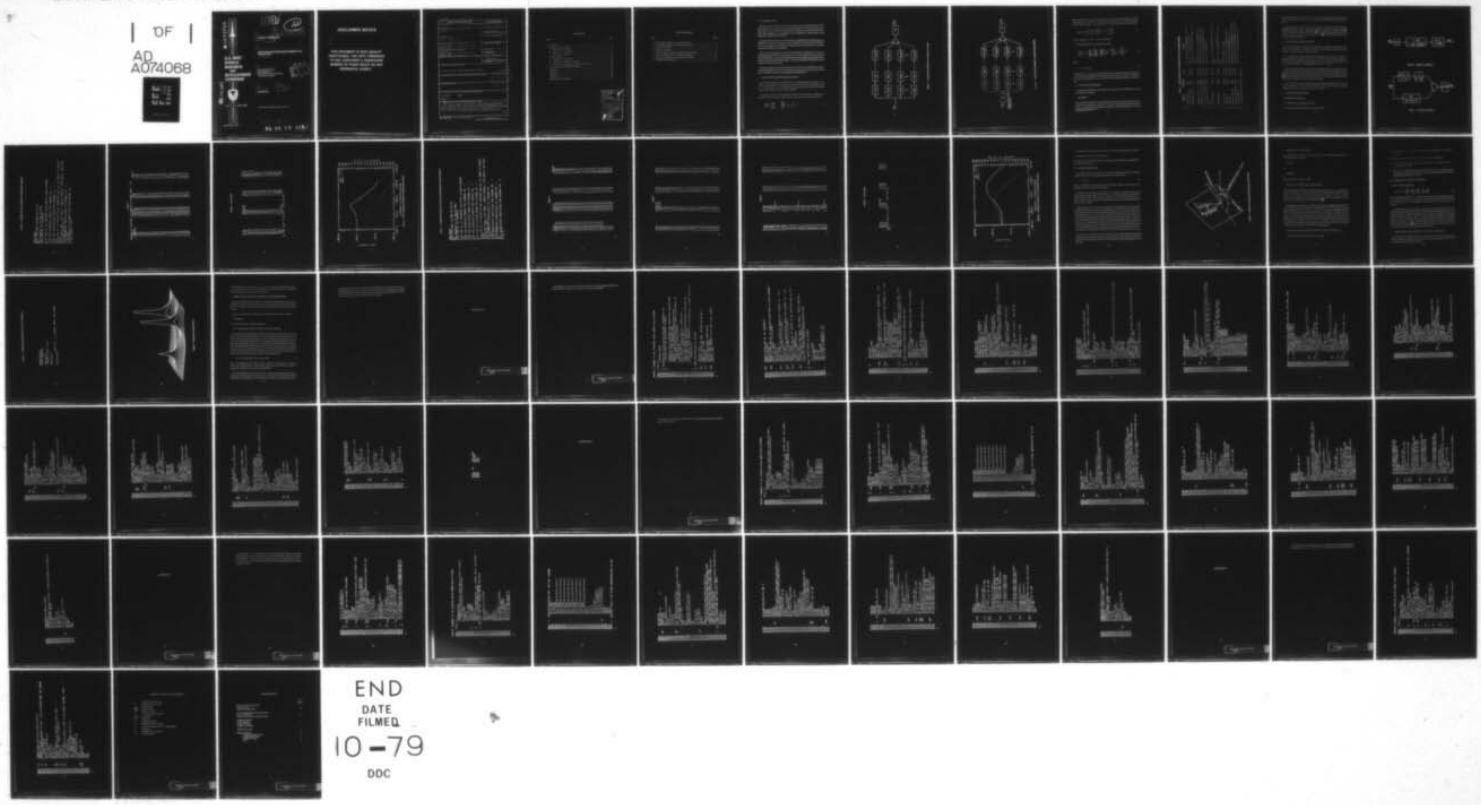
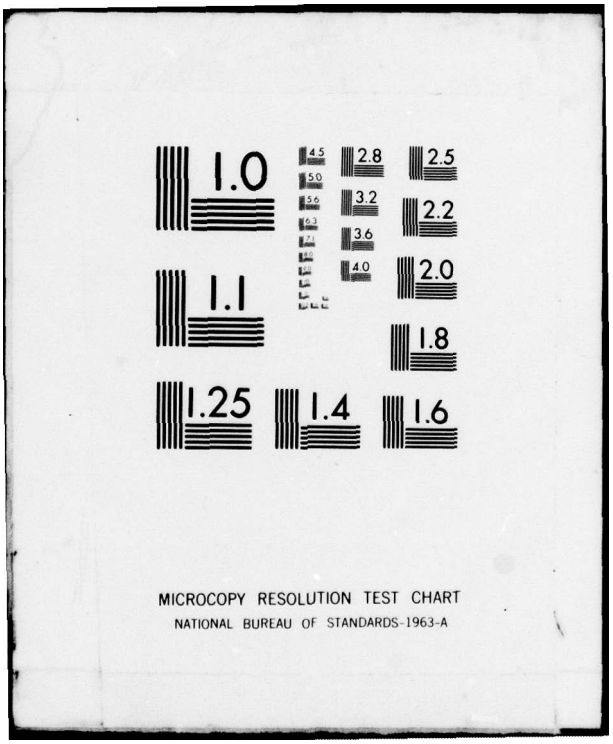
AD-A074 068 ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX.(U)
JUN 79 M C PITRUZZELLO, J R MITCHELL

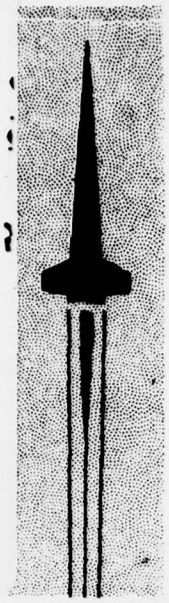
UNCLASSIFIED DRDMI-T-79-65 NL

| OF |
AD
A074068



END
DATE
FILMED
10-79
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



LEVEL II

12

14 IRDMI-T-79-65

9 TECHNICAL REPORT T-79-65

U.S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND

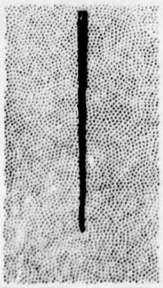
6 ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX

10 Michael C. Pitruzzello
Jerrel R. Mitchell
Guidance and Control Directorate
Technology Laboratory

11 26 June 1979

12 48 p.

Redstone Arsenal, Alabama 35809



Approved for Public Release; Distribution unlimited.

393427

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER T-79-65	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerrel R. Mitchell Michael C. Pitruzzello		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS ATTN: Commander US Missile Research and Development Command DRSMI-TG (R&D) Redstone Arsenal, AL 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ATTN: Commander US Missile Research and Development Command DRSMI-TI (R&D) Redstone Arsenal, AL 35809		12. REPORT DATE 26 June 79 13. NUMBER OF PAGES 77
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

M-II SYSTEM FRESP

FORTRAN IV

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains the description of four computer codes developed for the HP-21 MX minicomputer system housed in the Control System Branch of the Guidance and Control Directorate of the US Army Missile Research and Development Command (MIRADCOM). These codes (programs) were developed to aid in analyzing fire control systems being developed or proposed by the US Army.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

Section		Page
1.	Introduction	3
2.	Frequency Reponse Program	3
	A. Data Input for FRESP	6
	B. Illustrative Uses of FRESP	8
	C. Limitations of FRESP	8
3.	3-D Plot Program	20
	A. Using PLOT3 and PLOTX	22
	B. Example Use of PLOT3 or PLOTX	23
	C. Limitations and Comments on PLOT3 and PLOTX	23
4.	Directory Read and Modification Program	26
	Appendix A	29
	Appendix B	45
	Appendix C	57
	Appendix D	69
	Abbreviations and Symbols	75

ILLUSTRATIONS

Figure Page

1. General Block Diagram for Continuous Analysis 4
2. General Block Diagram for Sampled-Data Analysis 5
3. System for Example 1 9
4. System for Example 2 9
5. Plot of Frequency Response of Sampled-Data Example 13
6. Frequency Response for the Continuous Example 19
7. Pictorial for Defining the LOS Angle and Tilt Angle 21
8. Example Plot Using PLOTX 25

1. INTRODUCTION

This report contains the description of four computer codes developed for the HP-21MX minicomputer system housed in the Control System Branch of the Guidance and Control Directorate of the US Army Missile Research and Development Command (MIRADCOM). These codes (programs) were developed to aid in analyzing fire control systems being developed or proposed by the US Army.

The names of the symbolic versions of the programs are $FRESP, $PLOT3, $PLOTX, and $DREAD. The names of the binary versions are FRESP, PLOT3, PLOTX and DREAD. Uses of these programs are described in the following sections, and listings of the symbolic versions are contained in the appendices.

Both the symbolic and binary version of FRESP, PLOT3, and DREAD are stored on the flexible disc labeled SYSCPY. The symbolic version of PLOTX ($PLOTX) is stored on the M-II SYSTEM disc. It was stored here because the user must supply the subroutine DATA and then recompile and relocate the program. The libraries on the M-II SYSTEM disc must be searched in the relocation process.

All binary programs must be executed under the control of the M-II SYSTEM contained on the M-II SYSTEM disc. Even if relocated properly, none of the codes can be executed under an M-I SYSTEM since they were developed using FORTRAN IV.

2. FREQUENCY RESPONSE PROGRAM

The program FRESP was developed to aid in computing frequency responses of continuous or sampled-data control systems. This program is stored on the disc labeled M-II and can be executed under control of the M-II System contained on the same disc. The symbolic version of this program is stored in the file $FRESP.

The general continuous and sampled-data systems for which FRESP is applicable are, respectively, shown in *Figures 1* and *2*. For the continuous case, the program computes

$$\frac{C(s)}{R(s)} = \left[\sum_{\ell=1}^{L} \prod_{i=1}^{I(k)} G_{ji}(s) \right] e^{-ds} \qquad (1)$$

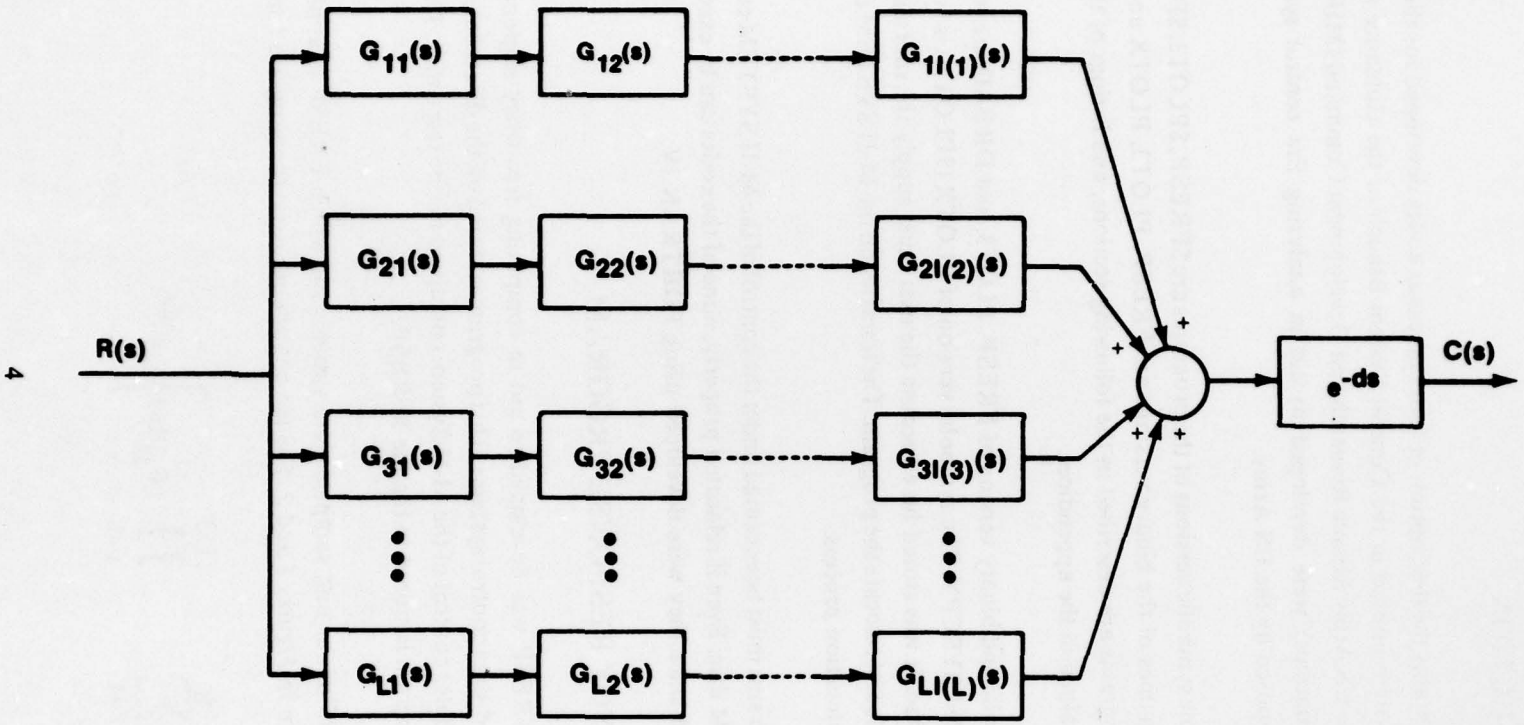


Figure 1. General block diagram for continuous analysis.

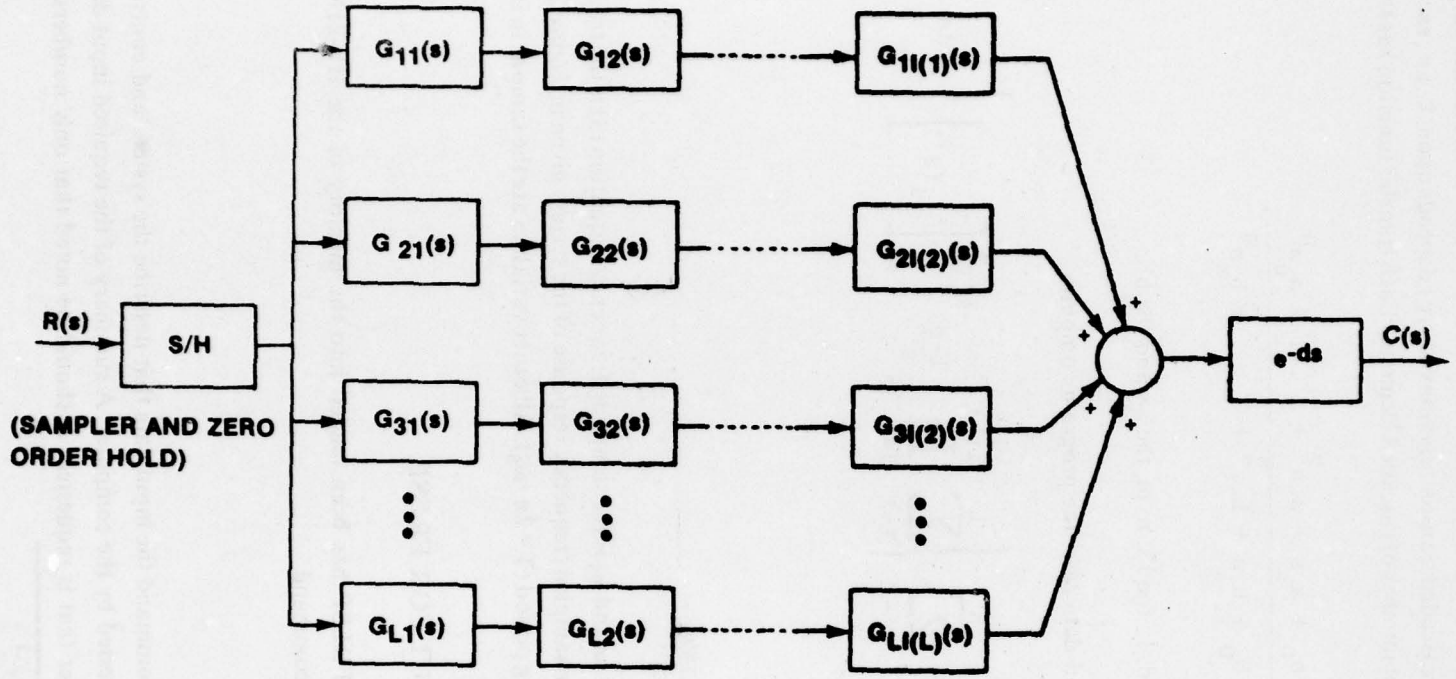


Figure 2. General block diagram for sampled-data analysis.

where $s=j\omega$.* There are L forward paths between the input, R(s), and the output, C(s). Each forward path has I cascaded transfer functions where I depends upon ℓ, i.e., each forward path can have a different number of cascaded elements. Each transfer function has the general form

$$G_{ki}(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_n s^n}{b_0 + b_1 s + b_2 s^2 + \ldots + b_m s^m} . \tag{2}$$

The user specifies d, L, the I's, n, m, the a's and the b's.

For the sampled-data case, the program computes

$$\frac{C^*(s)}{R^*(s)} = \frac{1}{T} \sum_{k=-K}^{K} \left\{ \sum_{\ell=1}^{L} \prod_{i=1}^{I(k)} \left[\frac{1-e^{-s'T}}{s'} \right] \left[G_{ji}(s') \right] \right\} e^{-ds} \tag{3}$$

where

$$s' = s + jk\omega_s . \tag{4}$$

In Equation (4), $s = j\omega$ and ω_s is the sampling frequency. Equation (3) is the truncated infinite series version of the sampled frequency response of the system given in *Figure 2*. In Equation (3) T is the sampling period ($T = 2\pi/\omega_s$). All other variables are the same as in the continuous case.

A. DATA INPUT FOR FRESP

Assuming that FRESP has been loaded into the memory of the HP-21MX, it can be executed with the command

ON, FRESP

After issuing this command the input data that describe the system and control the flow of execution are requested by the computer. A summary of the required input data is given in *Table 1* in the order that is requested. It should be noted that only numbers 1 and 15 are

*The symbol j represents $\sqrt{-1}$.

TABLE 1. DEFINITIONS OF REQUIRED INPUTS FOR USING FRESP.

NO.	QUANTITY READ-IN	PROGRAM NAME OF QUANTITY	FORMAT	COMMENTS
1.	Type of Analysis	MODE	A2	CR for continuous on SR for sampled-data. Default is CR.
2.	No. of Parallel Paths	KCHNL	free field	Corresponds to L in Equations (1) and (3).
3.	Transfer functions/path	NUMC	free field	Array for storing number of cascaded transfer functions in each path.
4.	Numerator Orders of T.F.'s	NRATOR	free field	Two-dimensional array for storing numerator orders.
5.	Denominator Order of T.F.'s	NDENOM	free field	Two-dimensional array for storing denominator orders.
6.	No. Points to be Completed	FREQPT	free field	Number of frequency response points to be computed.
7.	No. of Terms to be used in SR.	NOTSIR	free field	Corresponds to K in Equation (3).
8.	Starting Frequency	FSTART	free field	Frequency at which the computation is to begin (units in rad/sec).
9.	Ending Frequency	FRQEND	free field	Frequency at which the computation is to end (units in rad/sec).
10.	Sampling Period	SAMPRD	free field	Units are assumed to be seconds.
11.	Transport Lag	DLFUNC	free field	Amount of time delay (d in Equations (1) and (3).
12.	Gain factor	GAIN	free field	The frequency response magnitude scale factor.
13.	Numerator Coefficients	XCOMN	free field	Two-dimensional array for storing numerator coefficients.*
14.	Denominator Coefficients	YCOMN	free field	Two-dimensional array for storing denominator coefficients.*
15.	Plotting Designator	M1	A2	YES for a plot. NO for no plot. Default is NO.

* The coefficients should be input in ascending order.

column dependent. When requested for these, the user must start his response in column one to be assured of a proper interpretation. It should also be noted that both of these have default values.

All other inputs are free field. If more than one number is required, the numbers should be separated by commas or blanks. Each line of input is terminated with a carriage return (CR). For a free field input, if a slash (/) followed by a (CR) is issued, the input can be continued to the next line. If a line of input has not been terminated, i.e., a (CR) given, the input of the line can be restarted by issuing a RUBOUT.

B. ILLUSTRATIVE USES OF FRESP

In order to illustrate the pragmatic use of FRESP, two examples are presented in this section. The first example is a single-forward path, sampled-data system, and the second example is a two-forward path continuous system.

(1) EXAMPLE 1. Consideration is given to the system shown in *Figure 3*. The goal is to compute the sampled frequency response when T = 0.02513 for $0.1 \leq \omega \leq 100$. Ten terms are to be used in the series and 50 frequency response points are to be computed and plotted. The session with the computer for accomplishing this is shown in *Table 2*.

On the first page of *Table 2* the dialogue with the computer for describing the system and controlling the computational flow is given. After the data is printed, the computer queries as to whether or not a plot is desired. In this case a plot was desired; therefore, YES followed by a (CR) was typed. (The YES is not shown in this case.) The screen was automatically erased, and the frequency response plot shown in *Figure 5* was made.

(2) EXAMPLE 2. In this example it is desired to compute and plot the frequency response of the system shown in *Figure 43* for $0.01 \leqslant \omega \leqslant 99.9$. The results of the session with the computer for this case are presented in *Table 3* and *Figure 6*.

C. LIMITATIONS OF FRESP

The limitations of FRESP are:

- Maximum of 5 forward paths (channels)
- Maximum of 10 transfer functions per forward path

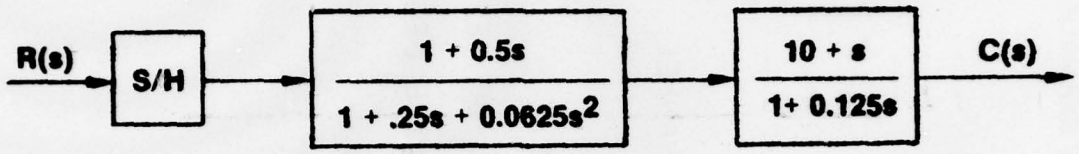


Figure 3. System for example 1.

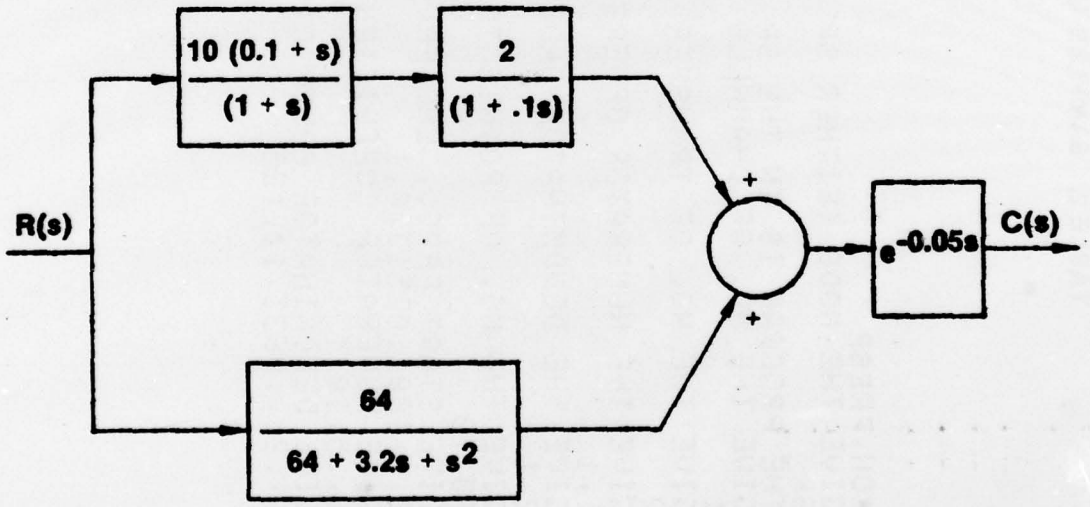


Figure 4. System for example 2.

TABLE 2. SAMPLED-DATA FREQUENCY RESPONSE EXAMPLE

```
*ON,FRESP
GIVE THE MODE (EITHER SR OR CR)
SR
THE PROGRAM IS IN THE SR MODE
GIVE THE NO. OF CHANNELS
1
GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL
2
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  1
1,1
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  1
2,1
GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF TERMS IN SERIES
50,10
GIVE START FREQ., END FREQ., SAMPLE PERIOD, TIME DELAY, AND GAIN
0.1,100,.02513,0,1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  1
1,.5,10,1
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  1
1,.25,.0625,1,.125
```

TABLE 2.

FRQEND IS CHANGED TO OMEGAS/2 SINCE MODE = SR

OMEGA	OMEGAW	MAG.	ANGLE
1.000E-01	.126E-02	10.0	1.21
.116	.145E-02	10.0	1.40
.134	.168E-02	10.0	1.62
.155	.194E-02	10.0	1.87
.179	.225E-02	10.0	2.16
.207	.260E-02	10.1	2.49
.239	.301E-02	10.1	2.87
.277	.348E-02	10.1	3.30
.320	.403E-02	10.2	3.80
.371	.466E-02	10.2	4.35
.429	.539E-02	10.3	4.98
.496	.623E-02	10.4	5.67
.573	.720E-02	10.5	6.43
.663	.833E-02	10.7	7.24
.767	.964E-02	10.9	8.07
.887	.111E-01	11.2	8.88
1.03	.129E-01	11.6	9.60
1.19	.149E-01	12.1	10.1
1.37	.173E-01	12.7	10.3
1.59	.200E-01	13.6	9.85
1.84	.231E-01	14.7	8.50
2.12	.267E-01	16.1	5.80
2.46	.309E-01	17.8	1.18
2.84	.357E-01	19.7	-6.03
3.29	.413E-01	21.2	-16.3
3.80	.478E-01	21.7	-29.3
4.40	.553E-01	20.7	-43.5
5.09	.640E-01	18.3	-56.5
5.88	.741E-01	15.5	-67.3

*

TABLE 2. (Concluded)

6.81	.857E-01	12.8	-75.5
7.87	.992E-01	10.6	-81.9
9.11	.115	8.75	-86.8
10.5	.133	7.27	-90.7
12.2	.154	6.07	-94.0
14.1	.179	5.10	-96.9
16.3	.208	4.31	-99.5
18.9	.241	3.66	-102.
21.8	.281	3.13	-105.
25.2	.328	2.68	-107.
29.2	.384	2.31	-111.
33.7	.451	1.99	-114.
39.0	.534	1.73	-118.
45.1	.637	1.51	-122.
52.2	.770	1.32	-128.
60.4	.948	1.17	-133.
69.8	1.20	1.04	-140.
80.8	1.61	.936	-148.
93.4	2.39	.859	-157.
108.	4.63	.809	-168.
125.	-.191E+06	.790	180.

DO YOU WANT TO PLOT? (YES OR NO)

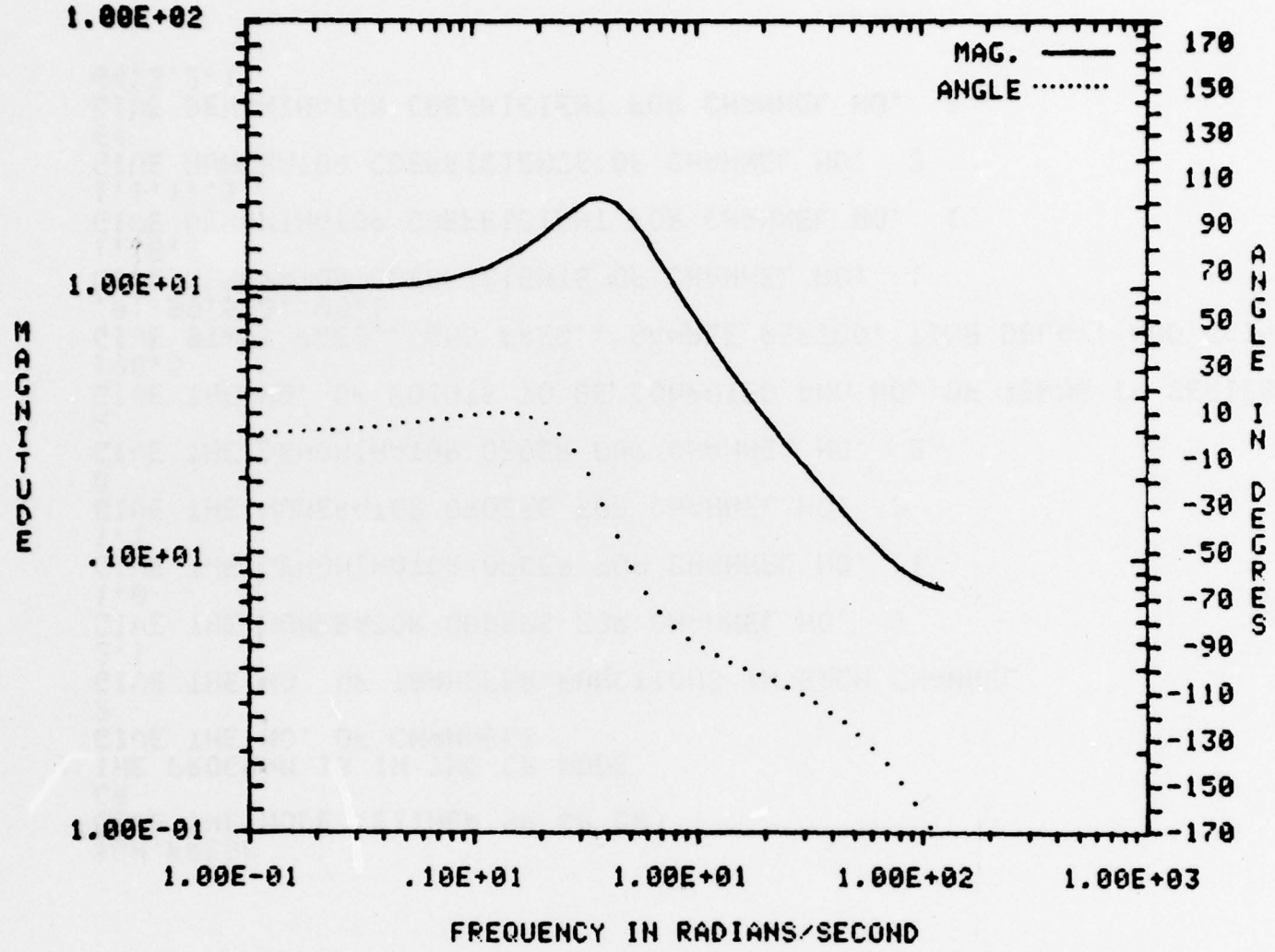


Figure 5. Plot of frequency response of sampled-data example.

TABLE 3. COMPUTER RESULTS FOR THE CONTINUOUS SYSTEM EXAMPLE

```
*ON,FRESP
GIVE THE MODE (EITHER SR OR CR)
CR
THE PROGRAM IS IN THE CR MODE
GIVE THE NO. OF CHANNELS
2
GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL
2,1
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  1
1,0
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  1
1,1
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  2
0
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  2
2
GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF TERMS IN SERIES
100,0
GIVE START FREQ., END FREQ., SAMPLE PERIOD, TIME DELAY, AND GAIN
.01,99.9,0,.05,1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  1
1,10,2
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  1
1,1,1,.1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  2
64
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  2
64,3.2,1
```

TABLE 3.

OMEGA	OMEGAW	MAG.	ANGLE
1.000E-02	.500E-02	3.01	3.35
.110E-01	.549E-02	3.01	3.68
.120E-01	.602E-02	3.01	4.04
.132E-01	.661E-02	3.01	4.43
.145E-01	.725E-02	3.02	4.86
.159E-01	.796E-02	3.02	5.33
.175E-01	.874E-02	3.02	5.84
.192E-01	.959E-02	3.03	6.40
.210E-01	.105E-01	3.03	7.01
.231E-01	.116E-01	3.04	7.68
.254E-01	.127E-01	3.05	8.42
.278E-01	.139E-01	3.06	9.21
.305E-01	.153E-01	3.07	10.1
.335E-01	.168E-01	3.08	11.0
.368E-01	.184E-01	3.10	12.0
.404E-01	.202E-01	3.12	13.2
.443E-01	.222E-01	3.14	14.4
.486E-01	.243E-01	3.17	15.6
.534E-01	.267E-01	3.20	17.0
.586E-01	.293E-01	3.24	18.5
.643E-01	.321E-01	3.28	20.0
.705E-01	.353E-01	3.34	21.7
.774E-01	.387E-01	3.40	23.4
.850E-01	.425E-01	3.48	25.2
.932E-01	.467E-01	3.57	27.1
.102	.512E-01	3.67	29.0
.112	.562E-01	3.80	30.9
.123	.617E-01	3.94	32.8
.135	.677E-01	4.10	34.6
.148	.744E-01	4.28	36.5

*

TABLE 3.

	.163	.816E-01	4.50	38.2
	.179	.896E-01	4.74	39.8
	.196	.984E-01	5.01	41.2
	.215	.108	5.31	42.5
	.236	.119	5.65	43.6
	.259	.130	6.03	44.5
	.285	.143	6.44	45.1
	.312	.158	6.90	45.5
	.343	.173	7.39	45.6
	.376	.190	7.92	45.4
	.413	.210	8.49	45.0
	.453	.231	9.10	44.3
	.497	.254	9.74	43.3
	.546	.280	10.4	42.1
	.599	.309	11.1	40.6
	.658	.341	11.8	38.9
	.722	.377	12.5	36.9
	.792	.418	13.2	34.7
	.869	.464	13.9	32.3
	.954	.517	14.6	29.8
	1.05	.577	15.3	27.0
	1.15	.647	15.9	24.2
	1.26	.730	16.5	21.3
	1.38	.829	17.0	18.2
	1.52	.950	17.5	15.1
	1.67	1.10	17.9	12.0
	1.83	1.30	18.3	8.76
	2.01	1.57	18.6	5.48
	2.20	1.97	18.8	2.16
	2.42	2.65	19.0	-1.23
	2.65	4.02	19.2	-4.68
	2.91	8.72	19.3	-8.21
*	3.20	-35.9	19.3	-11.8

TABLE 3.

	3.51	-5.38	19.4	-15.6
	3.85	-2.70	19.3	-19.4
	4.23	-1.66	19.3	-23.4
	4.64	-1.08	19.2	-27.6
	5.09	-.679	19.0	-32.1
	5.59	-.363	18.9	-36.8
	6.13	-.759E-01	18.7	-42.1
	6.73	.227	18.5	-48.3
	7.39	.615	18.0	-55.7
	8.11	1.29	16.8	-63.3
	8.90	3.69	15.3	-68.8
	9.76	-5.86	14.2	-72.9
	10.7	-1.33	13.4	-77.2
	11.8	-.427	12.7	-82.2
	12.9	.171	12.0	-87.8
	14.2	1.03	11.4	-93.8
	15.5	12.2	10.7	-100.
	17.1	-1.25	10.0	-107.
	18.7	-.634E-01	9.34	-114.
	20.5	1.14	8.69	-122.
	22.6	-3.48	8.06	-129.
	24.7	-.194	7.46	-138.
	27.2	1.61	6.88	-146.
	29.8	-1.04	6.34	-156.
	32.7	.761	5.83	-166.
	35.9	-1.25	5.35	-176.
	39.4	1.15	4.91	172.
	43.2	-.384	4.50	160.
	47.5	-5.82	4.12	147.
	52.1	1.29	3.77	132.
	57.2	.321	3.44	117.
	62.7	-.443E-01	3.15	99.8
*	68.9	-.128	2.87	81.5

TABLE 3. (Concluded)

```
   75.6              .876E-01          2.62              61.5
   82.9              .728              2.39              39.7
   91.0              25.0              2.18              15.9
   99.9              -.326             1.99             -10.1
DO YOU WANT TO PLOT?  (YES OR NO)
YES
```

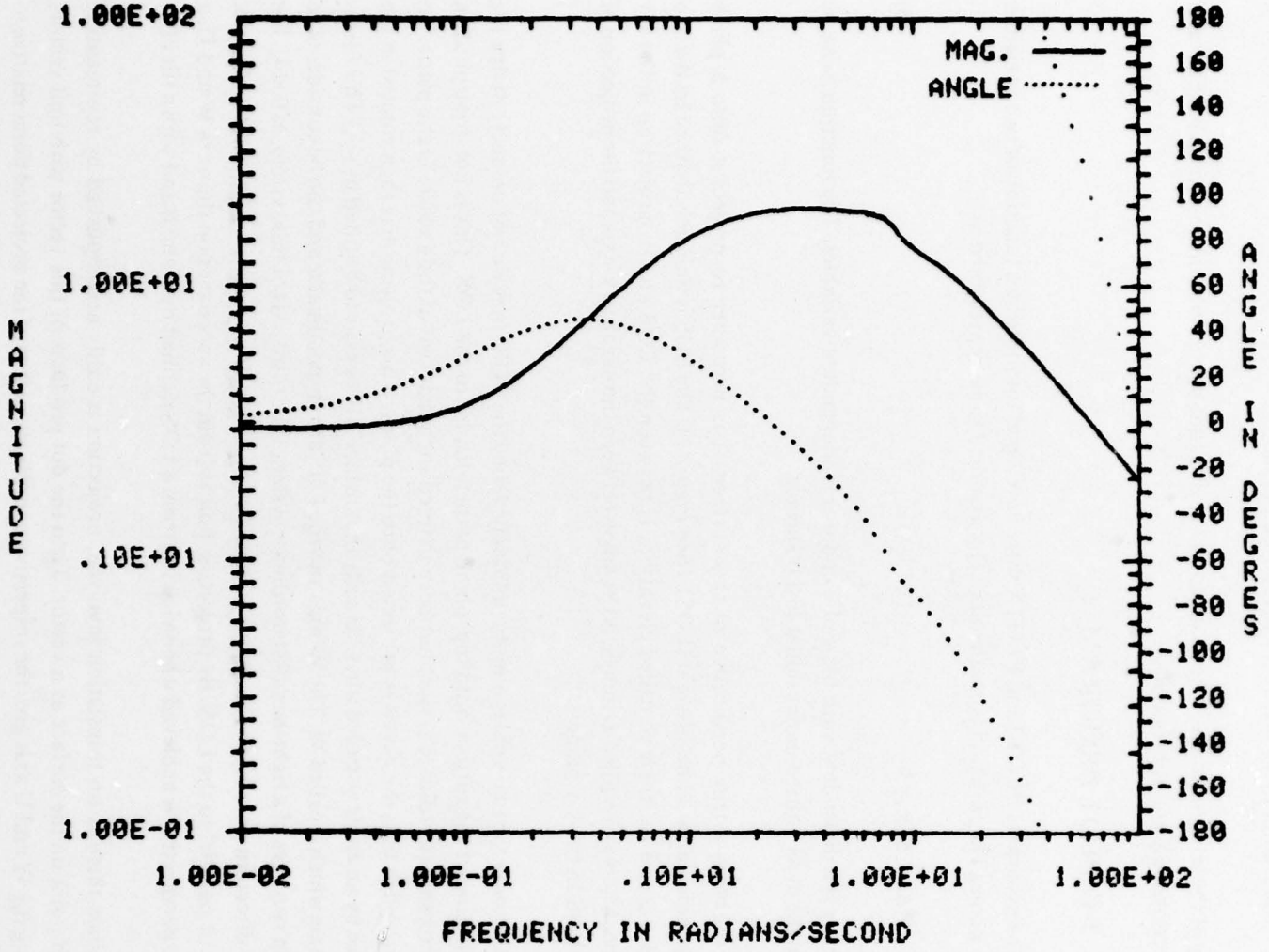


Figure 6. Frequency response for the continuous example.

- Maximum of 30 transfer function numerator or denominator coefficients per forward path

- Maximum of 100 frequency response points.

Any of these limitations can be changed by varying the appropriate dimensions in FRESP and the corresponding subroutines.

3. 3-D PLOT PROGRAM

The programs PLOT3 and PLOTX were developed for plotting a function of two variables in a plane. The function is mathematically assumed to be represented as

$$z = z(x,y) \tag{5}$$

where z is a dependent variable and x and y are independent variables. The function z can be thought of as a three-dimensional (3-D) surface.

To obtain a true perspective of this surface, it is necessary to project it onto a plane perpendicular to a line-of-sight (LOS). (See *Figure 7*). The LOS is uniquely defined by the two angles α and β. Angle β, called elevation, is between the LOS and its projection in the x-y plane, whereas α, called azimuth, is the angle between the negative y-axis and the projection of the LOS in the x-y plane.

Although the two angles α and β uniquely define the LOS, they do not uniquely define the projection of the surface onto the plane perpendicular to the LOS. This is not done until an orthogonal coordinate system has been defined such that two of the axes lie in the plane. To accomplish this, the z-axis is projected onto the plane. This projection is then rotated in the plane by an angle γ, called tilt. (Actually the surface will appear to be tilted by $-\gamma$.) This forms an axis which is called W. The W-axis and the LOS are perpendicular and can be considered to form two axes of a right-handed coordinate system. The third axis of this system, called U, lies in a direction parallel to the vector formed by crossing a unit vector parallel to W and a unit vector parallel to the LOS. By projecting points from the surface upon the axes W and U, a true perspective is achieved when viewing from a LOS defined by α and β and with a tilt of γ.

Since there is no translation involved, projection is easily accomplished by representing each point on the surface as a vector. Then the dot products of this vector with unit vectors along the W and U axis give the respective coordinate points of the projected point relative to these axes.

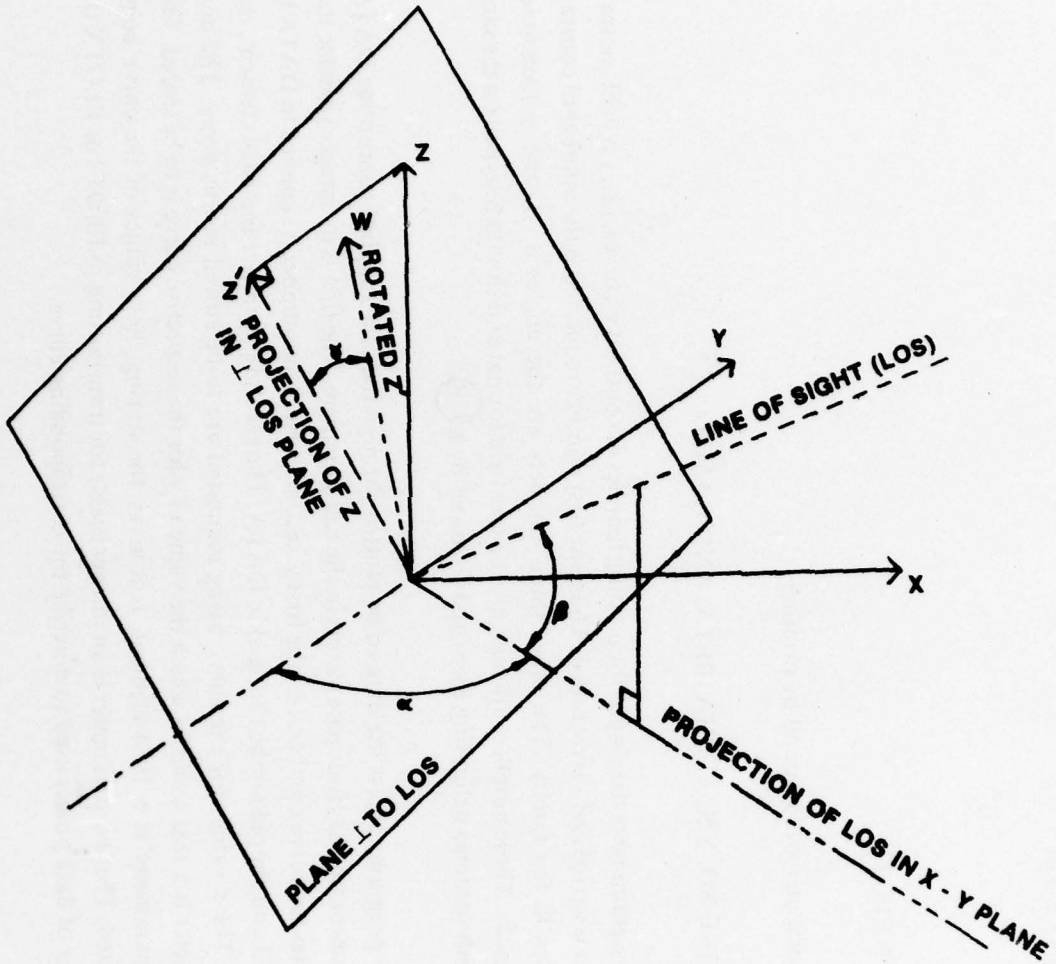


Figure 7. Pictorial for defining the LOS angle and tilt angle.

A. USING PLOT3 AND PLOTX

Assuming PLOT3 or PLOTX has been loaded into the HP-21MX, they can be run respectively with the commands

ON,PLOT3

and

ON,PLOTX.

The computer will respond by printing

GIVE NO. Y'S, ALPHA, BETA, AND GAMMA .

The programs were developed for plotting families of constant y-curves, i.e., y is held constant while x is varied and z is recorded. Thus, the first number requested is the number of constant y-curves in the family. The other three numbers are the angles α, β, and γ mentioned previously. These numbers are read using a free field format and should be typed on the same line with commas delimiting them and followed by a (CR).

The programs access the data to be plotted through the user supplied subroutine DATA. This subroutine has four parameters in the call statement. The first is an integer variable that indicates the curve number, i.e., the first y, second y, etc. This number is supplied to DATA by PLOT3 and should not be changed in DATA. The second parameter is a 1024 element, real array. The x values for the curve being requested are to be stored in this array. The next parameter is a real scalar in which the value of y for the respective curve is to be stored. The third parameter is a 1024 element, real array for storing the z values of the curve being requested. The last parameter is an integer scalar for transmitting to PLOT3 or PLOTX the number of data points used to describe the corresponding curve.

In essence the exchanges of data between PLOT3 or PLOTX and DATA are:

- PLOT3 or PLOTX gives DATA the curve number

- DATA returns the x's, a y value, the z's and the number of data points, i.e., the number of (z,x) pairs.

In developing the subrouting DATA the following rules should be adhered to:

- The x's must be either in ascending or descending order, i.e., random order of the x's will cause incorrect plotting.

- The curves are numbered starting with the lowest value of y if $-90° \leq \alpha \leq 90°$ and starting with the highest value of y if $90 < \alpha < 270$. (They are numbered starting with one and ending with the number of curves.)

B. EXAMPLE USE OF PLOT3 OR PLOTX

Consider the following function:

$$z_1(s) = 0.5 \left| \frac{s^2 + 4s + 53}{s^2 - 10s + 50} \right| \left| \frac{s}{s + 5} \right| \tag{6}$$

where s is the complex variable $s = x + jy$. Thus, the function z can be represented by Equation (5). It is desired to plot z for x on the range of $-10 \leq x \leq 10$ and y on the range $-9.25 \leq y \leq 10.75$. Twenty constant y-curves are to be plotted. For each y-curve 101 (z,x) pairs are to be generated.

The SUBROUTINE DATA presented in the listings of both PLOT3 and PLOTX was written for generating the data to be plotted. This subroutine was attached to both PLOT3 and PLOTX; both were compiled and relocated. The session on the computer for loading and executing PLOTX is shown in *Table 4*. Initially, PLOT3 was occupying the computer memory. It was "OFF-ed" and PLOTX was loaded. PLOTX was then run with the ON, PLOTX command and it immediately requested the number of y's, α, β, and γ. The line 2Ø, 15, 1Ø, Ø was typed followed by a (CR). The execution began and finally the plot shown in *Figure 8* was produced. Using PLOT3 the same results were obtained.

C. LIMITATIONS AND COMMENTS ON PLOT3 AND PLOTX

The programs PLOT3 and PLOTX are, in essence, the same. The major difference is that PLOTX requires about 2048 less words of computer memory than does PLOT3. PLOTX was specifically developed to obtain this savings.

TABLE 4. COMPUTER SESSION FOR THE PLOT EXAMPLE

```
*OF,PLOT3,8
PLOT3 ABORTED

*LO,PLOTX
 APLDR: DONE- PLOTX

*ON,PLOTX

          GIVE NO. Y'S, ALPHA, BETA, AND GAMMA
20,15,10,0
```

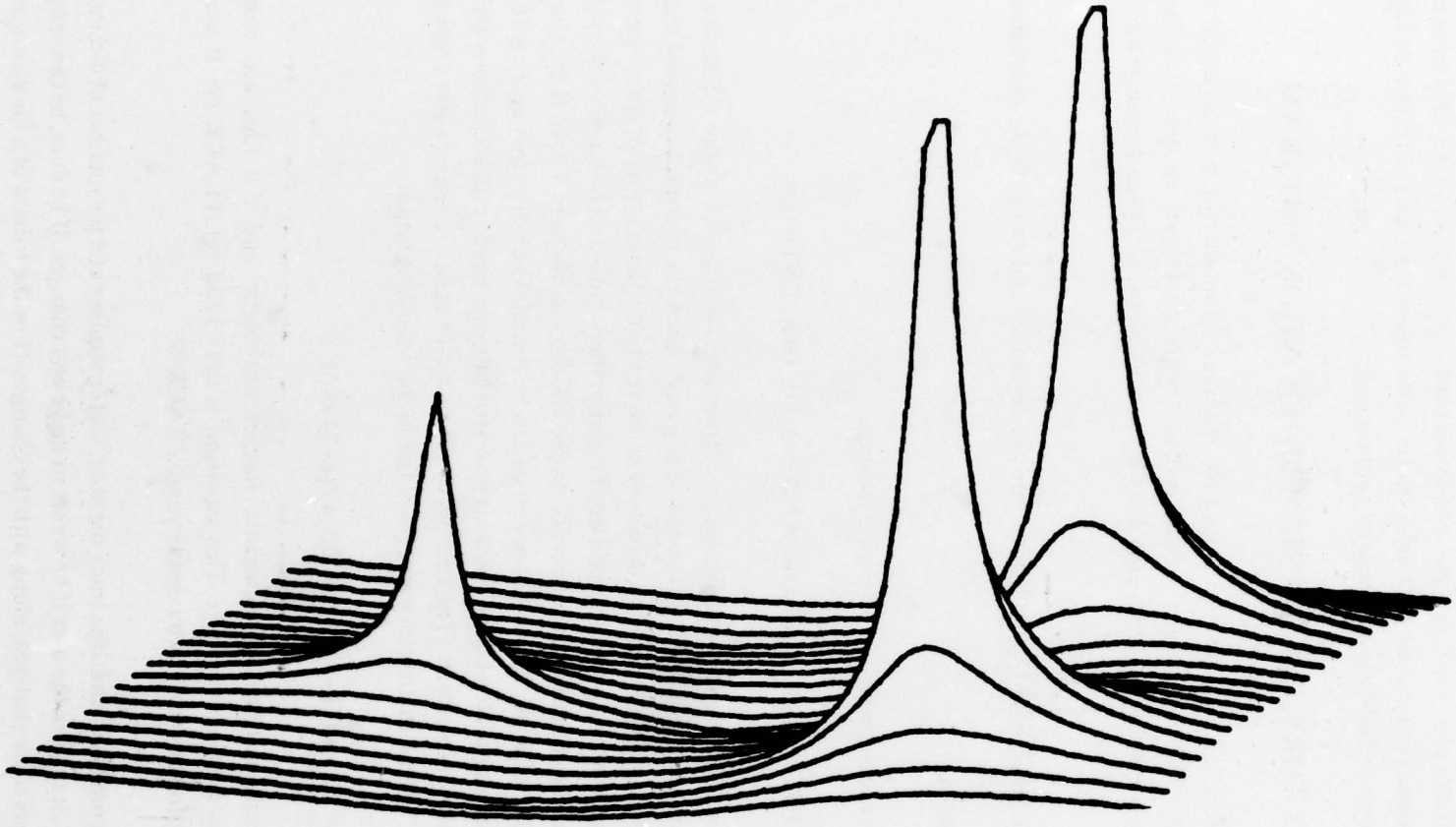


Figure 8. Example plot using PLOTX.

Both programs plot constant y-curves one at a time. For any y-curve the maximum number of (z,x) pairs must not exceed 1024 unless the dimensions on the z and x arrays are changed in the main program and in the associated subprograms.

4. DIRECTORY READ AND MODIFICATION PROGRAM

The program DREAD was developed for reading and modifying directories on the HP flexible discs. Such a program was needed to restore directories on discs that had been destroyed accidentally, e.g., changing discs without using the software dismount and mount routines.

In order to use DREAD, it is loaded into the computer and run with the command

```
*ON,DREAD
```

The computer then prints the following message

```
GIVE THE TRACK, SECTOR, AND NO. OF DIR. ENTRIES.
```

At this point the disc has been dismounted (from a software point of view). The disc can be removed from the drive and a different one inserted. Then the user must type the track and sector numbers of the part of the directory to be read and the number of directories in the sector to be viewed and possibly modified. These numbers should all be typed on the same line and separated by commas. The directory on the HP discs is stored in TRACK 66, starting at sector 0. In TRACK 66, 239 directory entries can be stored. Each directory entry is 16 words long. Each sector on TRACK 66 contains room for 8 directories except sector zero which has room for only 7 directories (The first 16 words of sector zero are used to store disc format information.) The Kth directory entry will fall in the following sector

$$IS = 14 * ((K-1)/8) - 60 * ((K-1)/40) \qquad (7)$$

where all computations are computer integer arithmetic and the files are numbered consecutively starting with *two*. This equation is only valid for TRACK 66. It would be unusual that the directory extended beyond TRACK 66.

After the computer reads the track number, sector number and the number of directories to be displayed, it queries the user if he wants to make any changes. If he does, he can specify the word numbers to be changed, along with the changes. (For the format of a file directory entry

see the RTE-M *Programmers Reference Manual*, Appendix F.) After a directory has been displayed and changes have been made, the changed directory entry is displayed. No changes have really been made at this point. When all the requested directory entries have been viewed, the process can be aborted by typing the number indicated.

APPENDIX A

The following is a file manager listing of the symbolic version of the program FRESP. The symbolic listing is contained on the SYSCPY disc in the file $FRESP.

```
$FRPOT T=00004 IS ON CR32760 USING 00045 BLKS R=5648

0001  FTN,L,T
0002        PROGRAM FRESP
0003  C
0004  C PROGRAM FOR CALCULATING FREQUENCY RESPONSE FOR
0005  C       SAMPLED-DATA OR CONTINUOUS SYSTEMS
0006  C
0007        DIMENSION ID(15),FREQRS(100),OMEGA(100),OMEGAW(100)
0008        DIMENSION AMAG(100),ANG(100)
0009        COMPLEX S,G,PHI,FREQRS,SIGMA,PHICAL,PHIDT
0010        COMMON XCOMN(5,30),YCOMN(5,30),NUMC(5),NRATOR(5,10)
0011       1  ,NDENOM(5,10),KNR(5),KDR(5)
0012        INTEGER FREQPT
0013        DATA IBLANK,ISDFR,ICDFR/2H  ,2HSR,2HCR/
0014        PHI(T,S)=(1.0-CEXP(-T*S))/S
0015        IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
0016        IYWRD(IPOS)=IXWRD(IPOS)+32
0017        POS(W,IH,IL,JH,JL)=FLOAT(IL)+((W-FLOAT(JL))/(FLOAT(JH-JL)))*
0018       1   FLOAT(IH-IL)
0019  C
0020  C  READ DESCRIPTION OF SYSTEM
0021  C
0022        WRITE(1,301)
0023  301   FORMAT("GIVE THE MODE (EITHER SR OR CR)")
0024        READ(1,101) MODE
0025  101   FORMAT(A2)
0026        IF(MODE.NE.ISDFR) MODE=ICDFR
0027        WRITE(1,200) MODE
0028  200   FORMAT("THE PROGRAM IS IN THE ",A2," MODE")
0029        WRITE(1,302)

*
```

```
0030  302    FORMAT("GIVE THE NO. OF CHANNELS")
0031         READ(1,*) KCHNL
0032         WRITE(1,303)
0033  303    FORMAT("GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL")
0034         READ(1,*) (NUMC(I),I=1,KCHNL)
0035         DO 2 I=1,KCHNL
0036         KNAT=NUMC(I)
0037         WRITE(1,304) I
0038  304    FORMAT("GIVE THE NUMERATOR ORDERS FOR CHANNEL NO. ",I2)
0039         READ(1,*) (NRATOR(I,J),J=1,KNAT)
0040         WRITE(1,305) I
0041  305    FORMAT("GIVE THE DENOMINATOR ORDER FOR CHANNEL NO. ",I2)
0042  2      READ(1,*) (NDENOM(I,J),J=1,KNAT)
0043         WRITE(1,306)
0044  306    FORMAT("GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF "
0045       1 "TERMS IN SERIES")
0046         READ(1,*) FREQPT,NOTSIR
0047         WRITE(1,307)
0048  307    FORMAT("GIVE START FREQ., END FREQ., SAMPLE PERIOD,"
0049       1 ," TIME DELAY, AND GAIN")
0050         READ(1,*) FSTART,FRQEND,SAMPRD,DLFUNC,GAIN
0051  C
0052  C  READ COEFFICIENTS OF BLOCKS
0053  C
0054         DO 3 I=1,KCHNL
0055         LAMP=NUMC(I)
0056         KNR(I)=0
0057         KDR(I)=0
0058         DO 4 J=1,LAMP
0059         KNR(I)=KNR(I)+NRATOR(I,J)+1
0060  4      KDR(I)=KDR(I)+NDENOM(I,J)+1

*
```

```
0061  3      CONTINUE
0062         DO 5 I=1,KCHNL
0063         LNC=KNR(I)
0064         LDC=KDR(I)
0065         WRITE(1,308) I
0066  308    FORMAT("GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO. ",I2)
0067         READ(1,*) (XCOMN(I,J),J=1,LNC)
0068         WRITE(1,309) I
0069  309    FORMAT("GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO. ",I2)
0070  5      READ(1,*) (YCOMN(I,J),J=1,LDC)
0071         IF(MODE.EQ.ICDFR)SAMPRD=1.0
0072         OMEGAS=3.14159265*2.0/SAMPRD
0073         IF(MODE.EQ.ICDFR) NOTSIR=0
0074         IF(MODE.EQ.ISDFR) FRQEND=OMEGAS/2.0
0075         IF(MODE.EQ.ISDFR) WRITE(1,310)
0076  310    FORMAT("FRQEND IS CHANGED TO OMEGAS/2 SINCE MODE = SR")
0077         FACTOR= (FRQEND/FSTART)**(1.0/(FLOAT(FREQPT)-1.0))
0078  C
0079  C   START FREQUENCY RESPONSE COMPUTATION LOOP
0080  C
0081         WRITE(1,212)
0082  212    FORMAT(3X,"OMEGA",10X,"OMEGAW",9X,"MAG.",11X,"ANGLE")
0083         OMEGA(1)=FSTART
0084         KOUNT=0
0085  10     KOUNT=KOUNT+1
0086         I=KOUNT
0087         IF(I.GT.100)GO TO 11
0088         IF(OMEGA(I).GT.FRQEND+FRQEND/1.0E+4) GO TO 11
0089         SIGMA=CMPLX(0.,0.)
0090         NNN=2*NOTSIR+1

*
```

```
0091            DO 7 NN=1,NNN
0092            N=-NOTSIR+NN-1
0093            S=CMPLX(0.,OMEGA(I)+N*OMEGAS)
0094            IF(MODE.EQ.ICDFR) GO TO 12
0095            PHICAL=PHI(SAMPRD,S)
0096    12      IF(MODE.EQ.ICDFR) PHICAL=1.0
0097            PHIDT=PHICAL*CEXP(-DLFUNC*S)*GAIN
0098            CALL EVAL(S,G,KCHNL)
0099    7       SIGMA=SIGMA+PHIDT*G
0100            FREQRS(KOUNT)=(1.0/SAMPRD)*SIGMA
0101            OMEGAW(I)=TAN(OMEGA(I)*SAMPRD/2.0)
0102            XMAG=CABS(FREQRS(I))
0103            XANG=ATAN2(AIMAG(FREQRS(I)),REAL(FREQRS(I)))*57.295779265
0104            WRITE(1,213) OMEGA(I),OMEGAW(I),XMAG,XANG
0105            AMAG(I)=XMAG
0106            ANG(I)=XANG
0107    213     FORMAT(G10.3,5X,G10.3,5X,G10.3,5X,G10.3)
0108            OMEGA(I+1)= OMEGA(I)*FACTOR
0109            GO TO 10
0110    11      WRITE(1,500)
0111    500     FORMAT("DO YOU WANT TO PLOT?  (YES OR NO)")
0112            READ(1,501) M1
0113    501     FORMAT(A2)
0114            IF(M1.EQ.54505B)GO TO 503
0115            STOP
0116    503     M1=15414B
0117            WRITE(1,501) M1
0118            I=I-1
0119            NPTS=I
0120            XMAX=AMAX(I,OMEGA)
0121            XMIN=AMIN(I,OMEGA)
0122            YMAX=AMAX(I,AMAG)

*
```

```
0123          YMIN=AMIN(I,AMAG)
0124          ZMAX=AMAX(I,ANG)
0125          ZMIN=AMIN(I,ANG)
0126    C     PROGRAM PLOT
0127    C     READ(1,*)YMAX,YMIN,XMAX,XMIN,ZMAX,ZMIN
0128    C     READ(1,*)NPTS
0129    C     READ(1,*) (AMAG(I),ANG(I),OMEGA(I),I=1,NPTS)
0130          IENT=29
0131          IEXT=31
0132    C  DEFINE CORNERS OF GRAPH
0133          IX1=IXWRD(185)
0134          IY1=IYWRD(94)
0135          IX2=IX1
0136          IY2=IYWRD(778)
0137          IX3=IXWRD(909)
0138          IY3=IY2
0139          IX4=IX3
0140          IY4=IY1
0141    C  FORM BORDER OF GRAPH
0142          WRITE(1,13)IENT,IY1,IX1,IY2,IX2,IY3,IX3,IY4,IX4,IY1,IX1,IEXT
0143    13    FORMAT(12A2)
0144          IX1=IXWRD(756)
0145          IY1=IYWRD(742)
0146          IX2=IXWRD(826)
0147          IY2=IYWRD(752)
0148          IY3=IY2
0149          IX3=IXWRD(882)
0150    C  PUT LEGEND ON GRAPH
0151          WRITE(1,15) IENT,IY1,IX1,IEXT,IENT,IY2,IX2,IY3,IX3,IEXT
0152    15    FORMAT(4A2,"MAG.",6A2)
0153          IX1=IXWRD(742)
0154          IY1=IYWRD(710)
0155          WRITE(1,16) IENT,IY1,IX1,IEXT

*
```

```
0156   16     FORMAT(4A2,"ANGLE")
0157          IY1=IYWRD(723)
0158          IX1=IXWRD(820)
0159          IY2=IY1
0160          IX2=IX1
0161          DO 19 I=1,9
0162          WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
0163          IX1=IXWRD(820+I*7)
0164   19     IX2=IX1
0165   C  PUT LABELS ON AXES
0166           WRITE(1,17)
0167   17     FORMAT(//////71X,"A"/71X,"N"/71X,"G"/"M",70X,"L"/"A",70X,
0168        .   1    "E"/"G"/"N",70X,"I"/"I",70X,"N"/"T"/"U",70X,"D"/"D",70X
0169        .   2    "E"/"E",70X,"G"/71X,"R"/71X,"E"/71X,"S")
0170          IY1=IYWRD(0)
0171          IX1=IXWRD(0)
0172          WRITE(1,18)IENT,IY1,IX1,IEXT
0173   18     FORMAT(4A2,25X,"FREQUENCY IN RADIANS/SECOND")
0174   C  DETERMINE NO. OF DECADES OF MAG. AND FREQ. AXES
0175          XMAX=XMAX-XMAX/100000.
0176          XMIN=XMIN+XMIN/100000.
0177          YMAX=YMAX-YMAX/100000.
0178          YMIN=YMIN+YMIN/100000.
0179          XMAX=ALOGT(XMAX)
0180          XMIN=ALOGT(XMIN)
0181          YMAX=ALOGT(YMAX)
0182          YMIN=ALOGT(YMIN)
0183          IYMAX=IFIX(YMAX)+1
0184          IXMAX=IFIX(XMAX)+1
0185          IYMIN=IFIX(YMIN)-1
0186          IXMIN=IFIX(XMIN)-1
0187          IDECX=IXMAX-IXMIN
0188          IDECY=IYMAX-IYMIN

*
```

```
0189  C  DETERMINE DECADE INCREMENT FOR MAG. AND FREQ. AXES
0190        ANCX=724./FLOAT(IDECX)
0191        ANCY=684./FLOAT(IDECY)
0192  C  PUT NUMBERS ON MAG. AXIS
0193        IY1=IYWRD(87)
0194        IX1=IXWRD(0)
0195        IUSE=IYMIN
0196        IDY1=IDECY+1
0197        DO 50 J=1,IDY1
0198        XNUM=10.**IUSE
0199        WRITE(1,40) IENT,IY1,IX1,IEXT,XNUM
0200  40    FORMAT(4A2,2X,E9.2)
0201        IUSE=IUSE+1
0202        APTS=FLOAT(J)*ANCY
0203  50    IY1=IYWRD(87+IFIX(APTS))
0204  C  PUT MAJOR TIC MARKS ON MAG. AXIS
0205        IY1=IYWRD(94)
0206        IX1=IXWRD(173)
0207        IY2=IY1
0208        IX2=IXWRD(185)
0209        DO 65 M=1,2
0210        DO 60 J=1,IDECY
0211        WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
0212  55    FORMAT(6A2)
0213        APTS=FLOAT(J)*ANCY
0214        IY1=IYWRD(94+IFIX(APTS))
0215  60    IY2=IY1
0216        IX1=IXWRD(897)
0217  65    IX2=IXWRD(909)
0218  C  PUT MINOR TIC MARKS ON MAG. AXIS
0219        IY1=IYWRD(94)
0220        IX1=IXWRD(177)

*
```

```
0221          IY2=IY1
0222          IX2=IXWRD(185)
0223          DO 75 M=1,2
0224          DO 70 J=1,IDECY
0225          DO 70 I=1,8
0226          Z=ALOGT(FLOAT(I+1))
0227          APTS=FLOAT(J-1)*ANCY+Z*ANCY
0228          IY1=IYWRD(94+IFIX(APTS))
0229          IY2=IY1
0230  70      WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
0231          IX1=IXWRD(901)
0232  75      IX2=IXWRD(909)
0233  C   PUT NUMBERS ON FREQUENCY AXIS
0234          IY1=IYWRD(47)
0235          IX1=IXWRD(75)
0236          IUSE=IXMIN
0237          IDX1=IDECX+1
0238          DO 80 J=1,IDX1
0239          XNUM=10.**IUSE
0240          IUSE=IUSE+1
0241          WRITE(1,40) IENT,IY1,IX1,IEXT,XNUM
0242          APTS=FLOAT(J)*ANCX
0243  80      IX1=IXWRD(75+IFIX(APTS))
0244  C   PUT MAJOR TIC MARKS ON FREQUENCY AXIS
0245          IY1=IYWRD(84)
0246          IX1=IXWRD(185)
0247          IY2=IYWRD(94)
0248          IX2=IX1
0249          DO 85 M=1,2
0250          DO 84 J=1,IDECX
0251          WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT

*
```

```
0252          APTS=FLOAT(J)*ANCX
0253          IX1=IXWRD(185+IFIX(APTS))
0254    84    IX2=IX1
0255          IY1=IYWRD(768)
0256    85    IY2=IYWRD(778)
0257    C  PUT MINOR TIC MARKS ON FREQUENCY AXIS
0258          IY1=IYWRD(87)
0259          IX1=IXWRD(185)
0260          IY2=IYWRD(94)
0261          IX2=IX1
0262          DO 95 M=1,2
0263          DO 93 J=1,IDECX
0264          DO 93 I=1,8
0265          Z=ALOGT(FLOAT(I+1))
0266          APTS=FLOAT(J-1)*ANCX+Z*ANCX
0267          IX1=IXWRD(185+IFIX(APTS))
0268          IX2=IX1
0269    93    WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
0270          IY1=IYWRD(771)
0271    95    IY2=IYWRD(778)
0272    C  PUT NUMBERS ON ANGLE AXIS
0273          IZMAX=IFIX(ZMAX/10.)+1
0274          IZMIN=IFIX(ZMIN/10.)-1
0275          ITEN=IZMAX-IZMIN
0276          ANCZ=684./FLOAT(ITEN)
0277          IY1=IYWRD(87)
0278          IX1=IXWRD(925)
0279          IUSE=IZMIN
0280          IDZ1=ITEN+1
0281          KJUMP=1
0282          DO 100 J=1,IDZ1
0283          INUM=10*IUSE

*
```

```
0284          IF(KJUMP.LT.0)GO TO 98
0285          WRITE(1,96) IENT,IY1,IX1,IEXT,INUM
0286    98    KJUMP=-KJUMP
0287    96    FORMAT(4A2,I4)
0288          IUSE=IUSE+1
0289          APTS=FLOAT(J)*ANCZ
0290    100   IY1=IYWRD(87+IFIX(APTS))
0291    C  PUT TIC MARKS ON ANGLE AXIS
0292          IX1=IXWRD(909)
0293          IX2=IXWRD(919)
0294          DO 110 M=1,2
0295          IY1=IYWRD(94)
0296          IY2=IY1
0297          DO 105 J=1,IDZ1
0298          WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
0299          APTS=FLOAT(J)*ANCZ
0300          IY1=IYWRD(94+IFIX(APTS))
0301    105   IY2=IY1
0302          IX1=IXWRD(185)
0303    110   IX2=IXWRD(195)
0304          IZMAX=IZMAX*10
0305          IZMIN=IZMIN*10
0306          IENT=29
0307          IEXT=31
0308          DO 49 I=1,NPTS
0309          W=ALOGT(AMAG(I))
0310          Y1=POS(W,778,94,IYMAX,IYMIN)
0311          W=ALOGT(OMEGA(I))
0312          X1=POS(W,909,185,IXMAX,IXMIN)
0313          IY1=IYWRD(IFIX(Y1))
0314          IX1=IXWRD(IFIX(X1))

*
```

```
0315          IF(I.LE.1)GO TO 45        *
0316          WRITE(1,30)IENT,IY1,IX1,IY2,IX2,IEXT
0317    30    FORMAT(6A2)
0318    45    IY2=IY1
0319          IX2=IX1
0320          Z1=POS(ANG(I),778,94,IZMAX,IZMIN)
0321          IZ1=IYWRD(IFIX(Z1))
0322    49    WRITE(1,30)IENT,IZ1,IX1,IZ1,IX1,IEXT
0323          READ(1,*)XYZ
0324          STOP
0325          END
0326          SUBROUTINE EVAL(S,G,KCHNL)
0327          DIMENSION CNUM(30),CDOM(30)
0328          COMPLEX S,G,FACT1,FACT2,RATIO
0329          COMMON XCOMN(5,30),YCOMN(5,30),NUMC(5),NRATOR(5,10),
0330         1  NDENOM(5,10),KNR(5),KDR(5)
0331          G=CMPLX(0.,0.)
0332          DO 10 I=1,KCHNL
0333          KCOMP=NUMC(I)
0334          KNOT=0
0335          KLOT=0
0336          RATIO=1.0
0337          DO 20 J=1,KCOMP
0338          NTR=NRATOR(I,J)+1
0339          NTD=NDENOM(I,J)+1
0340          DO 30 M=1,NTR
0341    30    CNUM(M)=XCOMN(I,M+KNOT)
0342          DO 40 M=1,NTD
0343    40    CDOM(M)=YCOMN(I,M+KLOT)
0344          KNOT=KNOT+NTR
0345          KLOT=KLOT+NTD
0346          K2=NTR-1
0347          K3=NTD-1
0348          CALL FUNC(CNUM,K2,S,FACT1)
```

```
0349            CALL FUNC(CDOM,K3,S,FACT2)
0350    20      RATIO=(FACT1/FACT2)*RATIO
0351    10      G=G+RATIO
0352            RETURN
0353            END
0354            SUBROUTINE FUNC(FW,K,X,SUM)
0355            DIMENSION FW(30)
0356            COMPLEX X,SUM
0357            SUM=CMPLX(FW(K+1),0.)
0358            IF(K.EQ.0) GO TO 20
0359            DO 10 II=1,K
0360            I=K-II+1
0361    10      SUM=SUM*X+CMPLX(FW(I),0.)
0362    20      RETURN
0363            END
0364            FUNCTION AMAX(N,X)
0365            DIMENSION X(100)
0366            AMAX=X(1)
0367            DO 10 I=1,N
0368            IF(X(I)-AMAX)10,10,5
0369    5       AMAX=X(I)
0370    10      CONTINUE
0371            RETURN
0372            END
0373            FUNCTION AMIN(N,X)
0374            DIMENSION X(100)
0375            AMIN=X(1)
0376            DO 10 I=1,N
0377            IF(X(I)-AMIN)5,10,10
0378    5       AMIN=X(I)

*
```

```
0379   10     CONTINUE
0380          RETURN
0381          END
0382          END$

:
```

APPENDIX B

The following is a file manager listing of PLOT3. It is contained on the disc SYSCPY and is located in the file $PLOT3.

```
$PLOT3 T=00004 IS ON CR32760 USING 00010 BLKS R=5652

0001   FTN,L,T
0002         PROGRAM PLOT4
0003         DIMENSION ZMAX(1024),ZMIN(1024),X(1024),Z(1024),
0004        1   U(3),W(3),Y1(50),P(8),W1(3),U1(8),S(3)
0005         RAD=57.2958
0006         KJUMP=0
0007         DO 5 I=1,1024
0008         ZMAX(I)=-5000.
0009   5     ZMIN(I)=5000.
0010         WRITE(1,2)
0011   2     FORMAT(/10X,36HGIVE NO. Y'S, ALPHA, BETA, AND GAMMA   )
0012         READ(1,*)N,ALPHA,BETA,GAMMA
0013   C  GET MAX. AND MIN. VALUES BEFORE ROTATION
0014         I=1
0015         CALL DATA(I,X,Y,Z,NPTS)
0016         XMX=X(1)
0017         XMN=X(1)
0018         YMX=Y
0019         YMN=Y
0020         ZMX=Z(1)
0021         ZMN=Z(1)
0022         DO 11 I=1,N
0023         CALL DATA(I,X,Y,Z,NPTS)
0024         IF(Y.GT.YMX)YMX=Y
0025         IF(Y.LT.YMN)YMN=Y
0026         DO 8 J=1,NPTS
0027         IF(X(J).GT.XMX)XMX=X(J)
0028         IF(X(J).LT.XMN)XMN=X(J)
0029         IF(Z(J).GT.ZMX)ZMX=Z(J)
0030   8     IF(Z(J).LT.ZMN)ZMN=Z(J)

*
```

```
0031   11     CONTINUE
0032   C  ASSIGN MAX. AND MIN VALUES FOR GRAPH
0033          KORGX=0
0034          KORGZ=0
0035          KXMAX=1000
0036          KZMAX=700
0037   C  COMPUTATION OF UNIT VECTOR ALONG LINE OF SIGHT (LOS)
0038          THETA=ALPHA-90.
0039          P(1)=COS(BETA/RAD)*COS(THETA/RAD)
0040          P(2)=COS(BETA/RAD)*SIN(THETA/RAD)
0041          P(3)=COS((90.-BETA)/RAD)
0042   C  PROJECTION OF Z-AXIS IN PERPENDICULAR PLANE OF LOS
0043   C  THE W-AXIS IS FORMED HERE
0044          W1(1)=-P(1)*P(3)
0045          W1(2)=-P(2)*P(3)
0046          W1(3)=1.-P(3)*P(3)
0047          WMAG=0.
0048          DO 12 I=1,3
0049   12     WMAG=WMAG+W1(I)*W1(I)
0050          WMAG=SQRT(WMAG)
0051          DO 14 I=1,3
0052   14     W1(I)=W1(I)/WMAG
0053   C  CROSS W1 AND LOS AXIS TO FORM U-AXIS
0054          U1(1)=W1(2)*P(3)-W1(3)*P(2)
0055          U1(2)=W1(3)*P(1)-W1(1)*P(3)
0056          U1(3)=W1(1)*P(2)-W1(2)*P(1)
0057   C  TILT W AND U AXES IN PLANE
0058          DO 15 I=1,3
0059          W(I)=SIN(GAMMA/RAD)*U1(I)+COS(GAMMA/RAD)*W1(I)
0060   15     U(I)=COS(GAMMA/RAD)*U1(I)-SIN(GAMMA/RAD)*W1(I)
0061   C  COMPUTE PROJECTED MAX. AND MIN. VALUES
0062          W1(1)=XMX

*
```

```
0063          W1(2)=YMX
0064          W1(3)=ZMX
0065          CALL DOTP(W1,U,W,P(1),U1(1))
0066          W1(3)=ZMN
0067          CALL DOTP(W1,U,W,P(2),U1(2))
0068          W1(2)=YMN
0069          CALL DOTP(W1,U,W,P(3),U1(3))
0070          W1(3)=ZMX
0071          CALL DOTP(W1,U,W,P(4),U1(4))
0072          W1(1)=XMN
0073          CALL DOTP(W1,U,W,P(5),U1(5))
0074          W1(3)=ZMN
0075          CALL DOTP(W1,U,W,P(6),U1(6))
0076          W1(2)=YMX
0077          CALL DOTP(W1,U,W,P(7),U1(7))
0078          W1(3)=ZMX
0079          CALL DOTP(W1,U,W,P(8),U1(8))
0080          A=P(1)
0081          B=P(1)
0082          C=U1(1)
0083          D=U1(1)
0084          DO 16 I=2,8
0085          IF(P(I).GT.A)A=P(I)
0086          IF(P(I).LT.B)B=P(I)
0087          IF(U1(I).GT.C)C=U1(I)
0088    16    IF(U1(I).LT.D)D=U1(I)
0089          XMX=A
0090          XMN=B
0091          ZMX=C
0092          ZMN=D
0093    C  PROJECT POINT INTO PLANE PERPENDICULAR TO LOS
0094          M1=15414B

*
```

```
0095            WRITE(1,98) M1
0096    98      FORMAT(A2)
0097            DO 7 J=1,N
0098            CALL DATA(J,X,Y,Z,NPTS)
0099            S(2)=Y
0100            DO 20 I=1,NPTS
0101            S(1)=X(I)
0102            S(3)=Z(I)
0103    20      CALL DOTP(S,U,W,X(I),Z(I))
0104    7       CALL PLOT3(XMX,XMN,ZMX,ZMN,Z,X,NPTS,ZMAX,ZMIN,
0105           1    KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
0106            READ(1,*) XYZ
0107            STOP
0108            END
0109            SUBROUTINE DOTP(S,U,V,X,Y)
0110            DIMENSION U(3),V(3),S(3)
0111            X=0.
0112            Y=0.
0113            DO 10 I=1,3
0114            X=X+S(I)*U(I)
0115    10      Y=Y+S(I)*V(I)
0116            RETURN
0117            END
0118            SUBROUTINE PLOT3(XMAX,XMIN,ZMX,ZMN,Z,X,
0119           1    NPTS,ZMAX,ZMIN,KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
0120            DIMENSION ZMAX(1024),ZMIN(1024),Z(50),X(50)
0121            IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
0122            IYWRD(IPOS)=IXWRD(IPOS)+32
0123            POS(W,IH,IL,AH,AL)=FLOAT(IL)+((W-AL)/(AH-AL))*FLOAT(IH-IL)
0124    C  FILL EDGE POINTS FOR MAX. AND MIN. ARRAY
0125            XS1=X(1)
0126            XE1=X(NPTS)
0127            ZS1=Z(1)

*
```

```
0128          ZE1=Z(NPTS)                    *
0129          IF(KJUMP.EQ.0)GO TO 100
0130          X2=XS2
0131          X1=XS1
0132          Z2=ZS2
0133          Z1=ZS1
0134          KSKIP=0
0135    10    CONTINUE
0136          KSTART=IFIX(POS(X1,KXMAX,KORGX,XMAX,XMIN))+1
0137          KEND=IFIX(POS(X2,KXMAX,KORGX,XMAX,XMIN))+1
0138          KPTS=KEND-KSTART
0139          IF(KPTS.EQ.0)GO TO 50
0140          KSGN=1
0141          IF(KPTS.LT.0)KSGN=-1
0142          KPTS=IABS(KPTS)
0143          DELX=(X2-X1)/FLOAT(KPTS)
0144          XV=X1
0145          SLOPE=(Z2-Z1)/(X2-X1)
0146          KEND=KPTS+1
0147          DO 20 I=1,KEND
0148          J=KSTART+KSGN*(I-1)
0149          ZV=SLOPE*(XV-X1)+Z1
0150          ZV=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
0151          IF(ZMAX(J).LT.ZV)ZMAX(J)=ZV
0152          IF(ZMIN(J).GT.ZV)ZMIN(J)=ZV
0153    20    XV=XV+DELX
0154    50    IF(KSKIP.EQ.1)GO TO 100
0155          X2=XE2
0156          X1=XE1
0157          Z2=ZE2
0158          Z1=ZE1
0159          KSKIP=1
0160          GO TO 10
0161   100    KJUMP=1
```

```
0162          XS2=XS1                    *
0163          XE2=XE1
0164          ZS2=ZS1
0165          ZE2=ZE1
0166    C  FILL-IN POINTS AND PLOT
0167          IENT=29
0168          IEXT=31
0169          I=0
0170    290   I=I+1
0171          IF(I.GE.NPTS)RETURN
0172          JX=0
0173          KSTART=IFIX(POS(X(I),KXMAX,KORGX,XMAX,XMIN))+1
0174          KEND=IFIX(POS(X(I+1),KXMAX,KORGX,XMAX,XMIN))+1
0175          NTOTAL=IABS(KEND-KSTART)
0176          IF(NTOTAL.EQ.0)RETURN
0177          DELX=(X(I+1)-X(I))/FLOAT(NTOTAL)
0178          KSGN=1
0179          IF(KSTART.GT.KEND)KSGN=-1
0180          K=KSTART-KSGN
0181          SLOPE=(Z(I+1)-Z(I))/(X(I+1)-X(I))
0182    300   XV=FLOAT(JX)*DELX+X(I)
0183          K=K+KSGN
0184          JX=JX+1
0185          IF(KSGN)301,301,302
0186    301   IF(XV.LT.X(I+1))GO TO 290
0187          GO TO 303
0188    302   IF(XV.GT.X(I+1))GO TO 290
0189    303   ZV=SLOPE*(XV-X(I))+Z(I)
0190          Z1=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
0191          IF(Z1.GE.ZMAX(K))GO TO 305
0192          IF(Z1.GT.ZMIN(K))GO TO 300
0193    305   X1=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0194          IZ1=IYWRD(IFIX(Z1))
0195          IX1=IXWRD(IFIX(X1+0.1))
```

```
0196            IF(Z1.GE.ZMAX(K))ZMAX(K)=Z1
0197            IF(Z1.LE.ZMIN(K))ZMIN(K)=Z1
0198    310     XV=FLOAT(JX)*DELX+X(I)
0199            K=K+KSGN
0200            JX=JX+1
0201            IF(KSGN)311,311,312
0202    311     IF(XV.LT.X(I+1))GO TO 320
0203            GO TO 314
0204    312     IF(XV.GT.X(I+1))GO TO 320
0205    314     KL=2
0206            ZV=SLOPE*(XV-X(I))+Z(I)
0207            Z3=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
0208            IF(Z3.GE.ZMAX(K))GO TO 315
0209            IF(Z3.GT.ZMIN(K))GO TO 325
0210    315     Z2=Z3
0211            X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0212            IF(Z2.GE.ZMAX(K))ZMAX(K)=Z2
0213            IF(Z2.LE.ZMIN(K))ZMIN(K)=Z2
0214            GO TO 310
0215    320     XV=X(I+1)
0216            ZV=Z(I+1)
0217            Z2=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
0218            X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0219            KL=1
0220    325     IZ2=IYWRD(IFIX(Z2))
0221            IX2=IXWRD(IFIX(X2+0.1))
0222            WRITE(1,328)IENT,IZ1,IX1,IZ2,IX2,IEXT
0223    328     FORMAT(6A2)
0224            IF(KL.EQ.1)GO TO 290.
0225            GO TO 300
0226            END
0227            SUBROUTINE DATA(II,X,Y,Z,NPTS)

*
```

```
0228            DIMENSION X(1024),Z(1024)
0229            COMPLEX S
0230            Z1(S)=.5*CABS((S**2+4.*S+53.)/(S**2-10.*S+50.)*(S/(S+5.)))
0231            IF(II.EQ.1)Y=-10.25
0232            Y=Y+1.0
0233            X(1)=-10.
0234            DO 10 I=1,101
0235            S=CMPLX(X(I),Y)
0236            Z(I)=Z1(S)
0237            IF(Z(I).GT.10.)Z(I)=10.
0238    10      X(I+1)=X(I)+.2
0239            NPTS=101
0240            RETURN
0241            END
0242            END$

:
```

APPENDIX C

The following is a file manager listing of the program PLOTX and the user supplied subroutine DATA. These are contained on the M-II SYSTEM disc, and both are located in the file $PLOTX. The subroutine DATA generates the data for the plot shown in *Figure 8*. For plotting other data, the user must substitute DATA with the appropriate version and then compile and relocate.

```
0029            IF(Z(J).GT.ZMX)ZMX=Z(J)
0030      8     IF(Z(J).LT.ZMN)ZMN=Z(J)
0031      11    CONTINUE
0032      C  ASSIGN MAX. AND MIN VALUES FOR GRAPH
0033            KORGX=0
0034            KORGZ=0
0035            KXMAX=1000
0036            KZMAX=700
0037      C  COMPUTATION OF UNIT VECTOR ALONG LINE OF SIGHT (LOS)
0038            THETA=ALPHA-90.
0039            P(1)=COS(BETA/RAD)*COS(THETA/RAD)
0040            P(2)=COS(BETA/RAD)*SIN(THETA/RAD)
0041            P(3)=COS((90.-BETA)/RAD)
0042      C  PROJECTION OF Z-AXIS IN PERPENDICULAR PLANE OF LOS
0043      C  THE W-AXIS IS FORMED HERE
0044            W1(1)=-P(1)*P(3)
0045            W1(2)=-P(2)*P(3)
0046            W1(3)=1.-P(3)*P(3)
0047            WMAG=0.
0048            DO 12 I=1,3
0049      12    WMAG=WMAG+W1(I)*W1(I)
0050            WMAG=SQRT(WMAG)
0051            DO 14 I=1,3
0052      14    W1(I)=W1(I)/WMAG
0053      C  CROSS W1 AND LOS AXIS TO FORM U-AXIS
0054            U1(1)=W1(2)*P(3)-W1(3)*P(2)
0055            U1(2)=W1(3)*P(1)-W1(1)*P(3)
0056            U1(3)=W1(1)*P(2)-W1(2)*P(1)
0057      C  TILT W AND U AXES IN PLANE
0058            DO 15 I=1,3
0059            W(I)=SIN(GAMMA/RAD)*U1(I)+COS(GAMMA/RAD)*W1(I)
0060      15    U(I)=COS(GAMMA/RAD)*U1(I)-SIN(GAMMA/RAD)*W1(I)
```

```
$PLOTX T=00004 IS ON CR32760 USING 00028 BLKS R=5652

0001  FTN,L,T
0002        PROGRAM PLOT4
0003        DIMENSION IZMX(1024),IZMN(1024),X(1024),Z(1024),
0004       1   U(3),W(3),P(8),W1(3),U1(8),S(3)
0005        RAD=57.2958
0006        KJUMP=0
0007        DO 5 I=1,1024
0008        IZMX(I)=-5000
0009  5     IZMN(I)=5000
0010        WRITE(1,2)
0011  2     FORMAT(/10X,36HGIVE NO. Y'S, ALPHA, BETA, AND GAMMA    )
0012        READ(1,*)N,ALPHA,BETA,GAMMA
0013  C  GET MAX. AND MIN. VALUES BEFORE ROTATION
0014        I=1
0015        CALL DATA(I,X,Y,Z,NPTS)
0016        XMX=X(1)
0017        XMN=X(1)
0018        YMX=Y
0019        YMN=Y
0020        ZMX=Z(1)
0021        ZMN=Z(1)
0022        DO 11 I=1,N
0023        CALL DATA(I,X,Y,Z,NPTS)
0024        IF(Y.GT.YMX)YMX=Y
0025        IF(Y.LT.YMN)YMN=Y
0026        DO 8 J=1,NPTS
0027        IF(X(J).GT.XMX)XMX=X(J)
0028        IF(X(J).LT.XMN)XMN=X(J)

*
```

```
0061   C  COMPUTE PROJECTED MAX. AND MIN. VALUES
0062          W1(1)=XMX
0063          W1(2)=YMX
0064          W1(3)=ZMX
0065          CALL DOTP(W1,U,W,P(1),U1(1))
0066          W1(3)=ZMN
0067          CALL DOTP(W1,U,W,P(2),U1(2))
0068          W1(2)=YMN
0069          CALL DOTP(W1,U,W,P(3),U1(3))
0070          W1(3)=ZMX
0071          CALL DOTP(W1,U,W,P(4),U1(4))
0072          W1(1)=XMN
0073          CALL DOTP(W1,U,W,P(5),U1(5))
0074          W1(3)=ZMN
0075          CALL DOTP(W1,U,W,P(6),U1(6))
0076          W1(2)=YMX
0077          CALL DOTP(W1,U,W,P(7),U1(7))
0078          W1(3)=ZMX
0079          CALL DOTP(W1,U,W,P(8),U1(8))
0080          A=P(1)
0081          B=P(1)
0082          C=U1(1)
0083          D=U1(1)
0084          DO 16 I=2,8
0085          IF(P(I).GT.A)A=P(I)
0086          IF(P(I).LT.B)B=P(I)
0087          IF(U1(I).GT.C)C=U1(I)
0088   16     IF(U1(I).LT.D)D=U1(I)
0089          XMX=A
0090          XMN=B
0091          ZMX=C
0092          ZMN=D
0093   C  PROJECT POINT INTO PLANE PERPENDICULAR TO LOS
       *
```

```
0094          M1=15414B
0095          WRITE(1,98) M1            *
0096    98    FORMAT(A2)
0097          DO 7 J=1,N
0098          CALL DATA(J,X,Y,Z,NPTS)
0099          S(2)=Y
0100          DO 20 I=1,NPTS
0101          S(1)=X(I)
0102          S(3)=Z(I)
0103    20    CALL DOTP(S,U,W,X(I),Z(I))
0104    7     CALL PLOT3(XMX,XMN,ZMX,ZMN,Z,X,NPTS,IZMX,IZMN,
0105         1    KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
0106          READ(1,*) XYZ
0107          STOP
0108          END
0109          SUBROUTINE DOTP(S,U,V,X,Y)
0110          DIMENSION U(3),V(3),S(3)
0111          X=0.
0112          Y=0.
0113          DO 10 I=1,3
0114          X=X+S(I)*U(I)
0115    10    Y=Y+S(I)*V(I)
0116          RETURN
0117          END
0118          SUBROUTINE PLOT3(XMAX,XMIN,ZMX,ZMN,Z,X,
0119         1    NPTS,IZMX,IZMN,KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
0120          DIMENSION IZMX(1024),IZMN(1024),Z(50),X(50)
0121          IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
0122          IYWRD(IPOS)=IXWRD(IPOS)+32
0123          POS(W,IH,IL,AH,AL)=FLOAT(IL)+((W-AL)/(AH-AL))*FLOAT(IH-IL)
0124    C  FILL EDGE POINTS FOR MAX. AND MIN. ARRAY
0125          XS1=X(1)
0126          XE1=X(NPTS)
0127          ZS1=Z(1)
0128          ZE1=Z(NPTS)
```

```
0129          IF(KJUMP.EQ.0)GO TO 100 *
0130          X2=XS2
0131          X1=XS1
0132          Z2=ZS2
0133          Z1=ZS1
0134          KSKIP=0
0135    10    CONTINUE
0136          KSTART=IFIX(POS(X1,KXMAX,KORGX,XMAX,XMIN))+1
0137          KEND=IFIX(POS(X2,KXMAX,KORGX,XMAX,XMIN))+1
0138          KPTS=KEND-KSTART
0139          IF(KPTS.EQ.0)GO TO 50
0140          KSGN=1
0141          IF(KPTS.LT.0)KSGN=-1
0142          KPTS=IABS(KPTS)
0143          DELX=(X2-X1)/FLOAT(KPTS)
0144          XV=X1
0145          SLOPE=(Z2-Z1)/(X2-X1)
0146          KEND=KPTS+1
0147          DO 20 I=1,KEND
0148          J=KSTART+KSGN*(I-1)
0149          ZV=SLOPE*(XV-X1)+Z1
0150          IZ1=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
0151          IF(IZMX(J).LT.IZ1)IZMX(J)=IZ1
0152          IF(IZMN(J).GT.IZ1)IZMN(J)=IZ1
0153    20    XV=XV+DELX
0154    50    IF(KSKIP.EQ.1)GO TO 100
0155          X2=XE2
0156          X1=XE1
0157          Z2=ZE2
0158          Z1=ZE1
0159          KSKIP=1
0160          GO TO 10
0161   100    KJUMP=1
0162          XS2=XS1
```

```
0163          XE2=XE1
0164          ZS2=ZS1
0165          ZE2=ZE1
0166    C  FILL-IN POINTS AND PLOT
0167          IENT=29
0168          IEXT=31
0169          I=0
0170    290   I=I+1
0171          IF(I.GE.NPTS)RETURN
0172          JX=0
0173          KSTART=IFIX(POS(X(I),KXMAX,KORGX,XMAX,XMIN))+1
0174          KEND=IFIX(POS(X(I+1),KXMAX,KORGX,XMAX,XMIN))+1
0175          NTOTAL=IABS(KEND-KSTART)
0176          IF(NTOTAL.EQ.0)RETURN
0177          DELX=(X(I+1)-X(I))/FLOAT(NTOTAL)
0178          KSGN=1
0179          IF(KSTART.GT.KEND)KSGN=-1
0180          K=KSTART-KSGN
0181          SLOPE=(Z(I+1)-Z(I))/(X(I+1)-X(I))
0182    300   XV=FLOAT(JX)*DELX+X(I)
0183          K=K+KSGN
0184          JX=JX+1
0185          IF(KSGN)301,301,302
0186    301   IF(XV.LT.X(I+1))GO TO 290
0187          GO TO 303
0188    302   IF(XV.GT.X(I+1))GO TO 290
0189    303   ZV=SLOPE*(XV-X(I))+Z(I)
0190          IZ1=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
0191          IF(IZ1.GE.IZMX(K))GO TO 305
0192          IF(IZ1.GT.IZMN(K))GO TO 300
0193    305   X1=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0194          IF(IZ1.GE.IZMX(K))IZMX(K)=IZ1
0195          IF(IZ1.LE.IZMN(K))IZMN(K)=IZ1

*
```

```
0196            X2=X1
0197            IZ2=IZ1
0198    310     XV=FLOAT(JX)*DELX+X(I)
0199            K=K+KSGN
0200            JX=JX+1
0201            IF(KSGN)311,311,312
0202    311     IF(XV.LT.X(I+1))GO TO 320
0203            GO TO 314
0204    312     IF(XV.GT.X(I+1))GO TO 320
0205    314     KL=2
0206            ZV=SLOPE*(XV-X(I))+Z(I)
0207            IZ3=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
0208            IF(IZ3.GE.IZMX(K))GO TO 315
0209            IF(IZ3.GT.IZMN(K))GO TO 325
0210    315     X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0211            IZ2=IZ3
0212            IF(IZ2.GE.IZMX(K))IZMX(K)=IZ2
0213            IF(IZ2.LE.IZMN(K))IZMN(K)=IZ2
0214            GO TO 310
0215    320     XV=X(I+1)
0216            ZV=Z(I+1)
0217            IZ2=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
0218            X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
0219            KL=1
0220    325     IX2=IXWRD(IFIX(X2+0.1))
0221            IZ2=IYWRD(IZ2)
0222            IX1=IXWRD(IFIX(X1+0.1))
0223            IZ1=IYWRD(IZ1)
0224            WRITE(1,328)IENT,IZ1,IX1,IZ2,IX2,IEXT
0225    328     FORMAT(6A2)
0226            IF(KL.EQ.1)GO TO 290
0227            GO TO 300
0228            END

*
```

```
0229            SUBROUTINE DATA(II,X,Y,Z,NPTS)
0230            DIMENSION X(1024),Z(1024)
0231            COMPLEX S
0232            Z1(S)=.5*CABS((S**2+4.*S+53.)/(S**2-10.*S+50.)*(S/(S+5.)))
0233            IF(II.EQ.1)Y=-10.25
0234            Y=Y+1.0
0235            X(1)=-10.
0236            DO 10 I=1,101
0237            S=CMPLX(X(I),Y)
0238            Z(I)=Z1(S)
0239            IF(Z(I).GT.10.)Z(I)=10.
0240    10      X(I+1)=X(I)+.2
0241            NPTS=101
0242            RETURN
0243            END
0244            END$

:
```

APPENDIX D

The following is a file management listing of the symbolic version of the program DREAD. This program is located on the disc SYSCPY. The symbolic version is in file $DREAD.

```
$DREAD T=00004 IS ON CR32760 USING 00007 BLKS R=0245

0001  FTN,L,T
0002        PROGRAM DREAD
0003        DIMENSION IBUF(128)
0004        CALL DCMC(1,-2,0)
0005        WRITE(1,6)
0006  6     FORMAT(47HGIVE THE TRACK, SECTOR, AND NO. OF DIR. ENTRIES
0007        READ(1,*) IT,IS,NFDS
0008        CALL DCMC(0,2,0)
0009  C  READ TRACT IT AND SECTOR IS
0010        CALL EXEC(1,2,IBUF,128,IT,IS)
0011  C  GET CONTENTS OF A AND B REG.'S
0012        CALL ABREG(IA,IB)
0013        IA=IAND(IA,377B)
0014        IF(IA)50,10,50
0015  10    IF(IS.NE.0)GO TO 11
0016        WRITE(1,9)
0017  9     FORMAT(/20X,10HDISC INFO. )
0018        WRITE(1,15) (IBUF(I),I=1,16)
0019  15    FORMAT(3A2,1X,13(I4,1X))
0020        KM=0
0021        GO TO 12
0022  11    KM=-16
0023  12    CONTINUE
0024        DO 20 J=1,NFDS
0025        WRITE(1,14) J
0026  14    FORMAT(/20X,8HDIR. NO.,I2)
0027        KS=J*16+1+KM
0028        KE=J*16+16+KM
0029        WRITE(1,16) (IBUF(I),I=KS,KE)

*
```

```
0030   16     FORMAT(3A2,1X,5(I4,1X),A2,7(1X,I4))
0031          WRITE(1,17) J
0032   17     FORMAT(/19HFOR CHANGES IN DIR. ,I2,1X,21HGIVE WRD. NO.,CHANG
0033         1   ,9HOR GIVE 0)
0034   19     READ(1,*) I1,I2
0035          IF(I1.LE.0)GO TO 22
0036          I1=I1+16*J
0037          IBUF(I1)=I2
0038          GO TO 19
0039   22     WRITE(1,23)
0040   23     FORMAT(/30HTHIS IS THE WAY IT WILL LOOK,  )
0041          WRITE(1,16) (IBUF(I),I=KS,KE)
0042   20     CONTINUE
0043          WRITE(1,25)
0044   25     FORMAT(/44HDO YOU REALLY WANT TO DO THIS? (0-YES, 1-NO)    )
0045          READ(1,*)KX
0046          IF(KX.NE.0)STOP
0047          CALL EXEC(2,2,IBUF,128,IT,IS)
0048          CALL ABREG(IA,IB)
0049          IA=IAND(IA,377B)
0050          IF(IA.NE.0)GO TO 50
0051          STOP
0052   50     WRITE(1,55) IA
0053   55     FORMAT(K5)
0054          STOP
0055          END
0056          END$

:
```

ABBREVIATIONS AND SYMBOLS

α	Azimuth angle of the LOS
β	Elevation angle of the LOS
$\lvert\ \rvert$	Absolute value
(CR)	Carriage Return
T.F.	Transfer Function
j	$\sqrt{-1}$ unless otherwise indicated
LOS	Line-of-sight
M-II	RTE-M II
ω	Frequency in rad/sec
ω_S	Sampling frequency in rad/sec
Π	Product if a large symbol; Pi if a small symbol
Σ	Summation
T	Sampling Period in seconds
3-D	Three-dimensional

DISTRIBUTION

	No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
US Army Materiel Systems Analysis Activity ATTN: DRXSY-MP Aberdeen Proving Ground, Maryland 21005	2
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, Illinois 60616	1
DRSMI-LP, Mr. Voigt	1
DRSMI-TBD (R&D)	3
-*TI (R&D) (Reference Copy)*	*1*
-TI (R&D) (Record Set)	1
-T (R&D), Dr. Kobler	1
-TGC	5